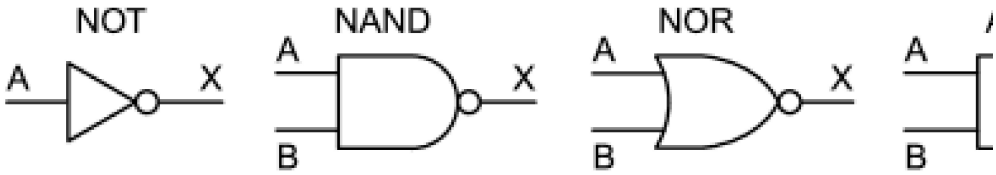
Stavebné prvky číslicových počítačov I

Obsah prednášky

- elementárne logické funkcie (AND, OR, NOT), opakovanie z PPI
- odvodené logické funkcie (NAND, NOR, XOR, NXOR), opakovanie z PPI
- schématické značky logických funkcií, opakovania z PPI
- sekvenčná logika (funkčnosť, schématická značka)
    - aynchrónny RS preklápací obvod (PO)
    - úrovňou riadený synchrónny D-PO
    - hranou riadený synchrónny D-PO
    - ďalšie PO (T-PO, JK-PO) a ich vzájomná transformácia z/na D-PO (len naznačenie súvislosti)
- nulovanie a nastavenie PO
- „set-up“ a „hold“ časy a ich význam v synchrónnych obvodoch, metastabilita
- rekonfigurovateľné obvody na báze FPGA
- principiálna schéma základného stavebného bloku FPGA – logický element (LE)
- princíp LUT tabuľky a jej využitia na realizáciu kombinačnej logiky (zdôraznenie odlišných požiadaviek na optimalizáciu implementácie logickej funkcie)
- kombinačná logika a jej optimalizácia (z pohľadu syntetizačných nástrojov)
- výstupné bloky FPGA obvodov
- vstavané bloky FPGA obvodov (násobičky, pamäťové bloky)
- vývoj konfigurácie FPGA obvodu, vývojové nástroje (väzba na cvičenia)
- hlavní výrobcovia FPGA obvodov

Logické funkcie (členy)



A	X
0	1
1	0

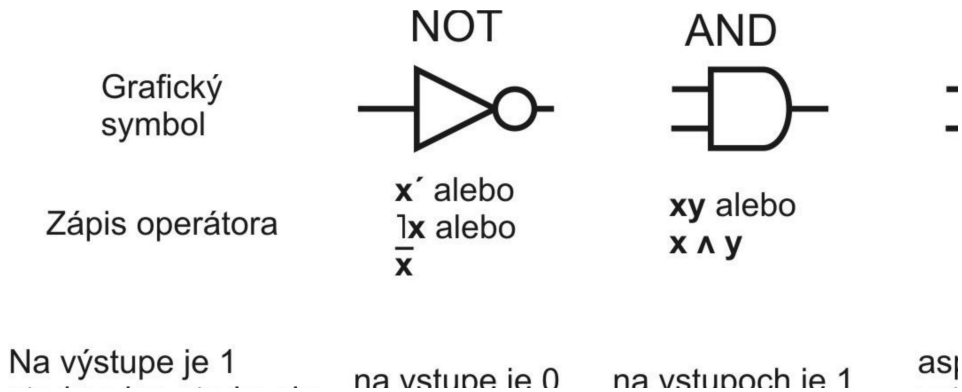
A	B	X
0	0	1
0	1	1

A	B	X
0	0	1
0	1	0

A
0
0

Symbolické označenie (ANSI) logických členov a ich pravdivostné tabuľky

AND, NOR, NOT - základné
NOR, NAND, XOR - odvodené



Rôzne v praxi využívané spôsoby zápisu logických obvodov

Metóda aritmetickej substitúcie

$z' = 1 - z$	NOT converted to a
xy	AND same as multi
	(when doing the alg

Príklad
S využitím aritmetických výrazov (aritmetickej substitúcie) overte platnosť podmienky

$$xyz \vee x' \vee y' \vee z' \equiv$$

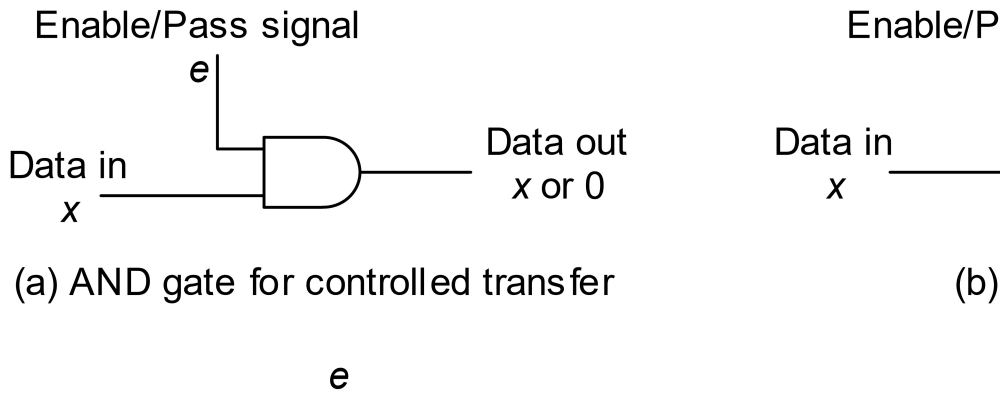
Riešenie

ľavá strana $= [xyz \vee x'] \vee [y' \vee z']$

$= [xyz + 1 - x - (1 - x)xyz] \vee [1 - y +$

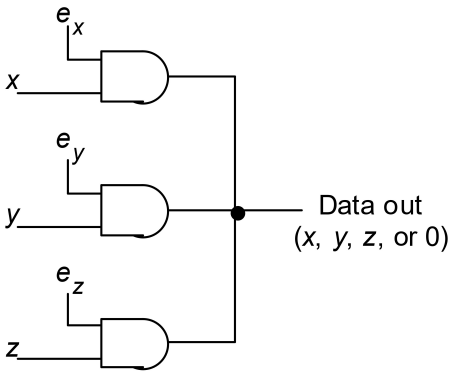
$= [xyz + 1 - x] \vee [1 - yz]$

Hradlá ako riadiace prvky (pre vytváranie zberníc)

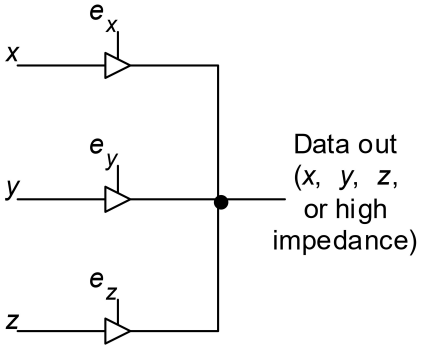


Troj-stavový „bufer“ (d) výstup je možné prepnúť do stavu vysokej impedancie (No data, označuje sa často symbolom Z).

Využitie riadiacich prvkov pri vytváraní zberníc



(a) Wired OR of product terms



(b) Wired OR of tristate outputs

Zákony (základné identity) boolovej algebry

Name of law	OR version	AND version
Identity	$x \vee 0 = x$	$x\ 1 = x$
One/Zero	$x \vee 1 = 1$	$x\ 0 = 0$
Idempotent	$x \vee x = x$	$x\ x = x$
Inverse	$x \vee x' = 1$	$x\ x' = 0$
Commutative	$x \vee y = y \vee x$	$x\ y = y\ x$
Associative	$(x \vee y) \vee z = x \vee (y \vee z)$	$(x\ y)\ z = x\ (y\ z)$
Distributive	$x \vee (y\ z) = (x \vee y)\ (x \vee z)$	$x\ (y \vee z) = (x\ y) \vee (x\ z)$
DeMorgan's	$(x \vee y)' = x'\ y'$	$(x\ y)' = x' \vee y'$

Kombinačný obvod je také zapojenie (elementárnych) logických obvodov, pre ktoré je výstupná logická hodnota daná len aktuálnymi logickými hodnotami signálov na jeho vstupe. Kombinačný obvod neobsahuje pamäť.

Vhodným prepojením elementárnych logických obvodov je možné vytvoriť **pamäťový element**, ktorý je základným stavebným blokom pamätí.

Sekvenčný obvod je také zapojenie (elementárnych) logických obvodov, ktoré obsahuje pamäť. Výstupná logická sekvenčného obvodu závisí nie len na **aktuálnych** vstupoch, ale aj na vstupoch v **predchádzajúcich** časových okamihoch.

Asynchrónny RS preklápací obvod (Flip-Flop) s využitím NOR (aktívny v log. 1)
http://www.learnabout-electronics.org/Digital/dig52.php

chyba v obrázku: v intervale "c" má byť Q=invQ=0

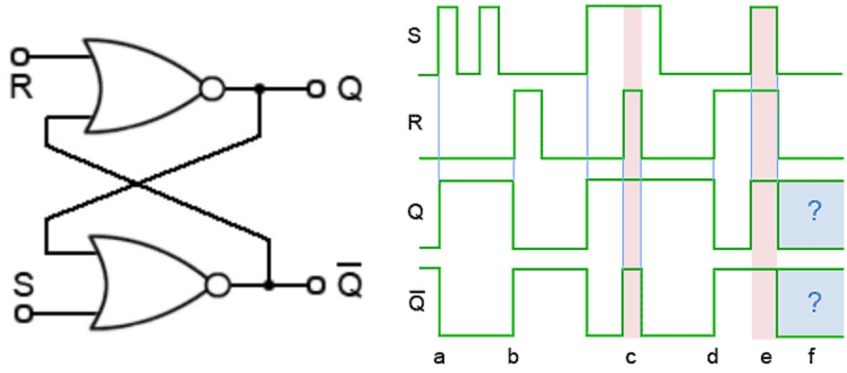


Table 5.2.2					
	R	S	Q	$\overline{Q}$	Comments
1.	0	1	1	0	Q is set to 1 by 1 on S
2.	0	0	1	0	No Change after set
3.	1	0	0	1	Q is reset to 0 by 1 on R
4.	0	0	0	1	No Change after reset
5.	1	1	0	0	Not allowed (both outputs at 0)
6.	0	0	?	?	If both inputs change from 11 to 00 outputs will be INDETERMINATE

S - vstup set (aktívny v 1)
R - vstup reset (aktívny v 1)

Asynchrónny RS preklápací obvod (Flip-Flop) s využitím NAND (aktívny v log. 0)

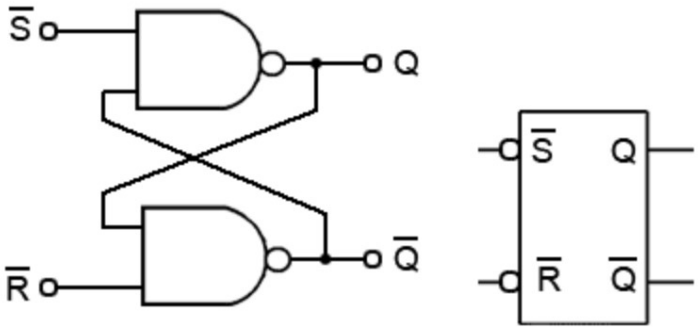
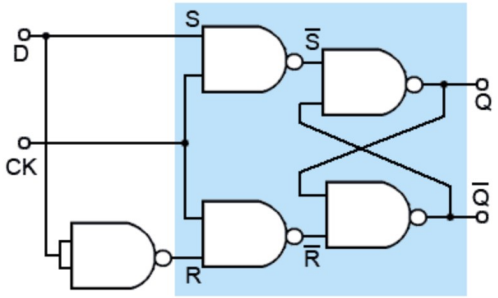


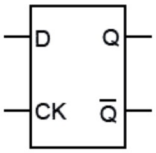
Table 5.2.1					
	$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	Comments
1.	0	1	1	0	Q is set to 1 by 0 on $\overline{S}$
2.	1	1	1	0	No change, (1 on Q is remembered)
3.	1	0	0	1	Q is reset to 0 by 0 on $\overline{R}$
4.	1	1	0	1	No change, (0 on Q is remembered)
5.	0	0	1	1	Both inputs at 0 – both outputs are at 1 (Non-allowed state)
6.	1	1	?	?	Inputs change from 0,0 to 1,1 together - outputs will be INDETERMINATE

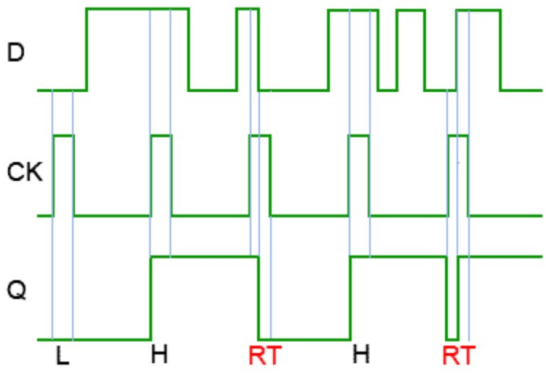
Úrovňou riadený synchrónny D preklápací obvod (Latch) s využitím NAND

http://www.learnabout-electronics.org/Digital/dig53.php

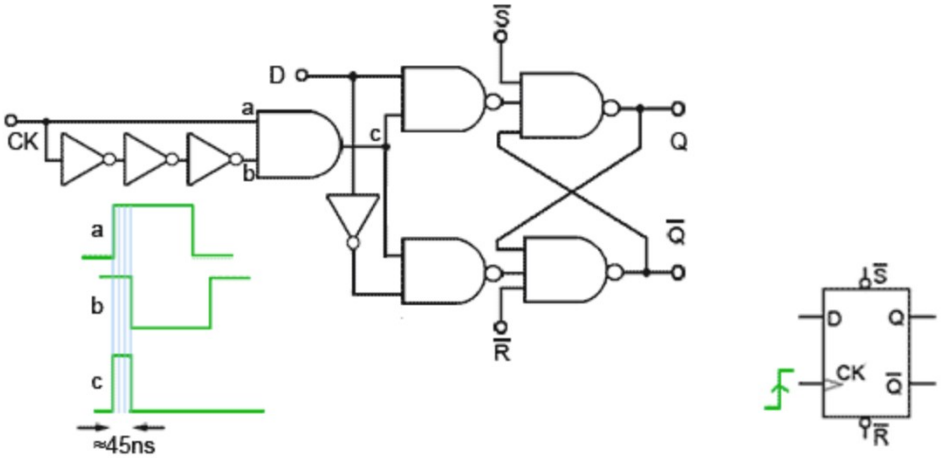


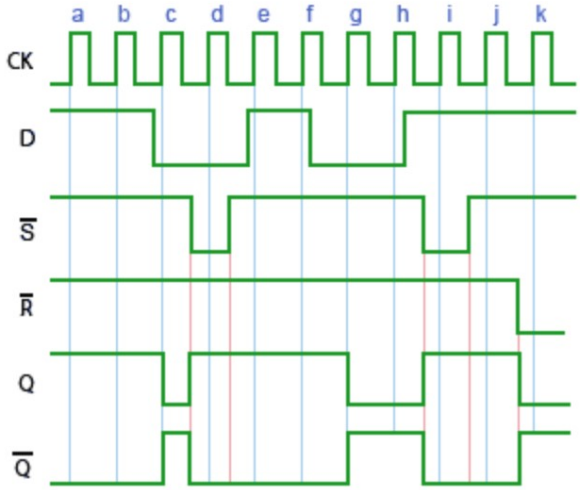
Inputs		Outputs	
CK	D	Q	$\overline{Q}$
0	X	No change	
1	0	0	1
1	1	1	0





Hranou riadený synchrónny D preklápací obvod (Flip-Flop)
s využitím NAND (jeden zo základných blokov v obvodoch FPGA)

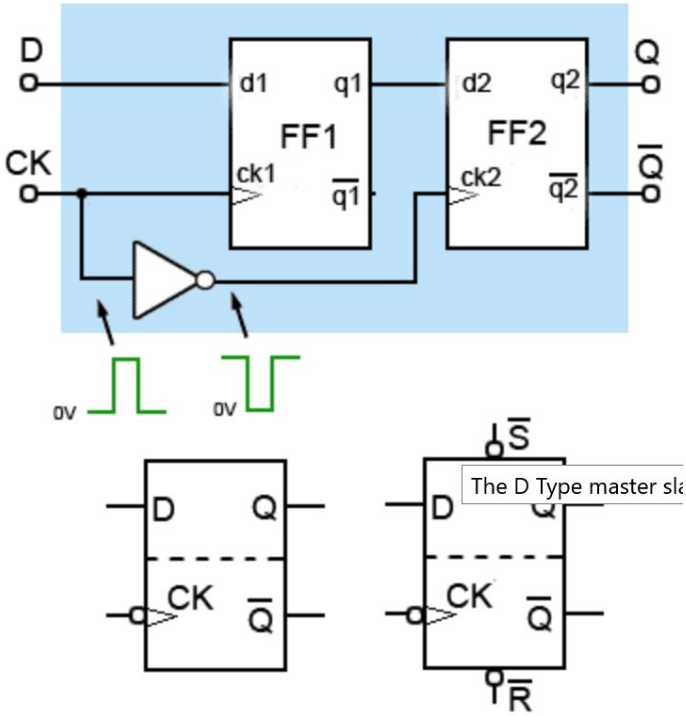


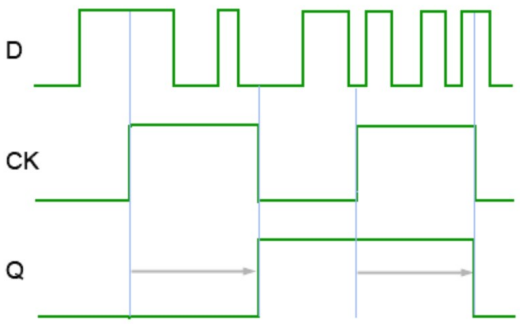


Poznámka
Uvedené zapojenie **je len principiálne**, v praxi sa používajú iné obvodové zapojenia (bez vytvárania úzkeho hodinového impulzu)

Obsahuje aj asynchrónny nulovací (R) a nastavovací (S) vstup.

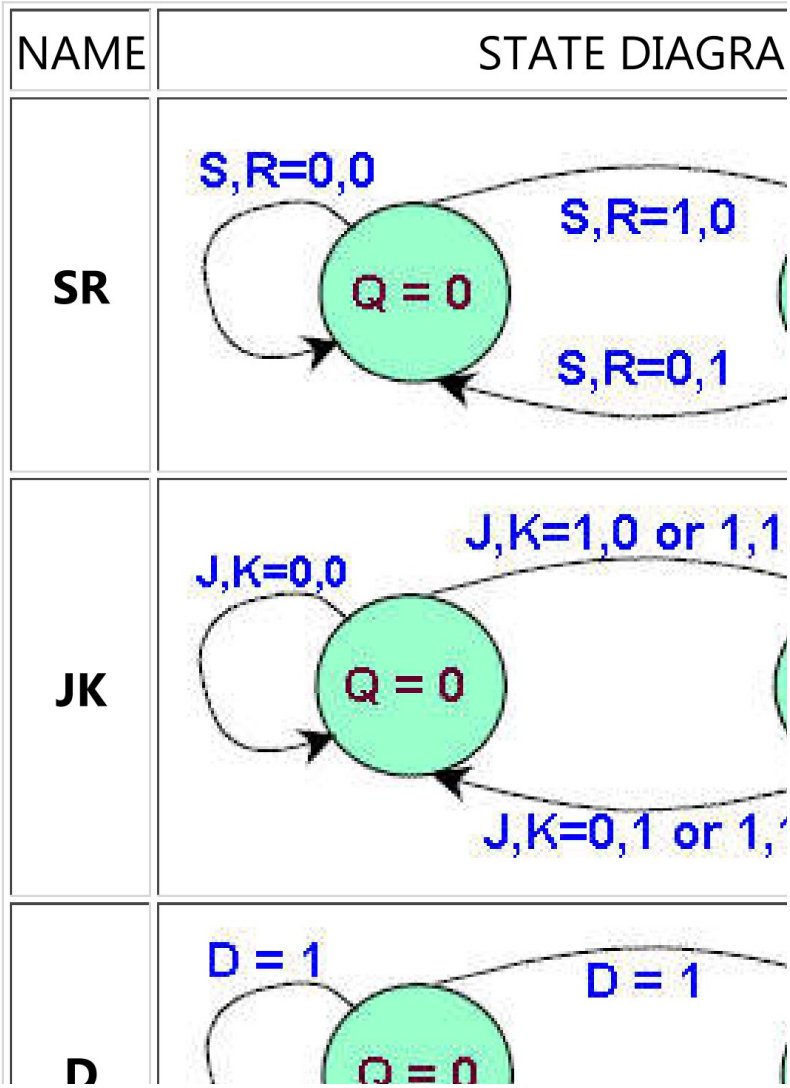
Hranou riadený synchrónny D preklápací obvod (Flip-Flop) typu master - slave





Oddeľuje vstup od výstupu, často sa využíva v digitálnych systémoch s **viac-fázovými hodinami**

Ďalšie preklápacie obvody (elementárne automaty) a ich stavové diagramy

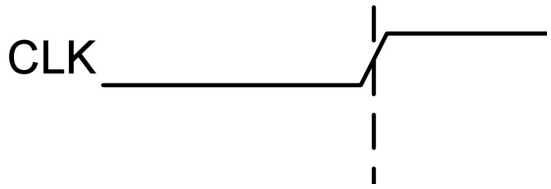


Elementárne automaty T a **JK** sa často používajú pri „ručnej“ syntéze digitálnych obvodov. Existujú aj vo forme integrovaných obvodov (IO) malej hustoty integrácie. Pomocou ktoréhokoľvek vyššie uvedeného automatu a dodatočnej kombinačnej logiky je možne vytvoriť všetky ostatné. Preto nám napr. v stavebných obvodoch (**logických elementoch - LE**) FPGA obvodov stačí len **synchrónny D preklápací obvod**.

Príklady konverzií
https://www.allaboutcircuits.com/technical-articles/conversion-of-flip-flops/

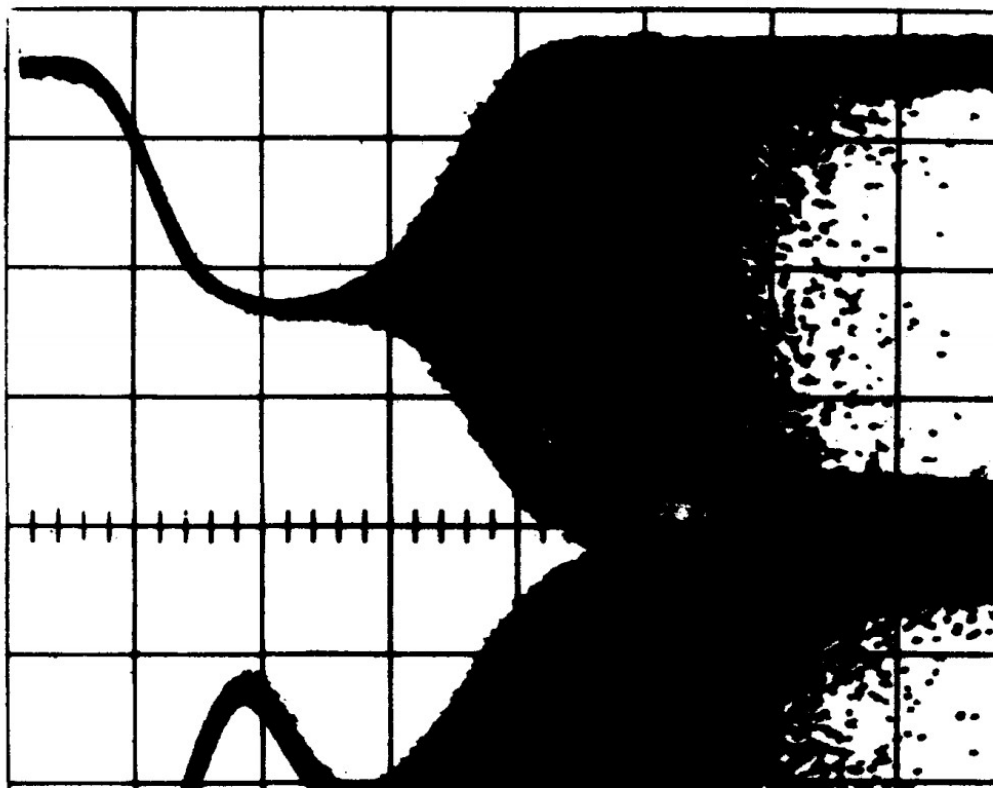
Časové obmedzenia vo vzťahu k hodinovým signálom

- Setup time: t_{setup} = time *before* the cloc be stable (i.e. not changing)
- Hold time: t_{hold} = time *after* the clock e stable
- Aperture time: t_a = time around clock e stable ($t_a = t_{setup} + t_{hold}$)



Metastabilita v číslicových obvodoch

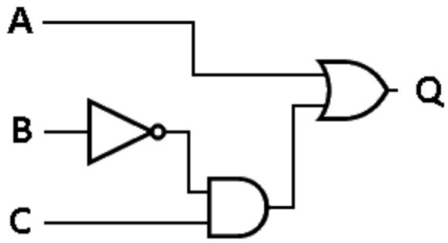
Nedodržanie setup a hold časov môže viesť k výraznému **predĺženiu** doby, počas ktorej bude logický obvod v nedefinovanom stave. **Zložitejšie sekvenčné obvody** (napr. zložitejšie konečné automaty) dokonca môžu **prestať fungovať!**

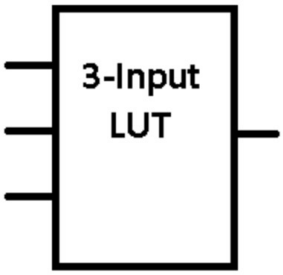


Princíp využitia LUT tabuľky na realizáciu kombinačnej logiky

https://www.nandland.com/articles/boolean-algebra-using-look-up-tables-lut.html

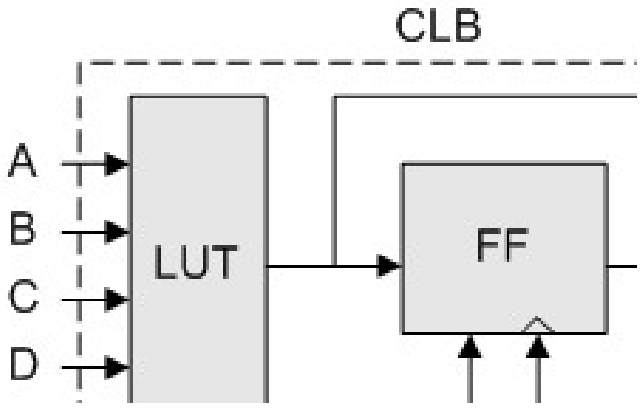
Truth Table - A + (C*B')			
Input A	Input B	Input C	Output Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



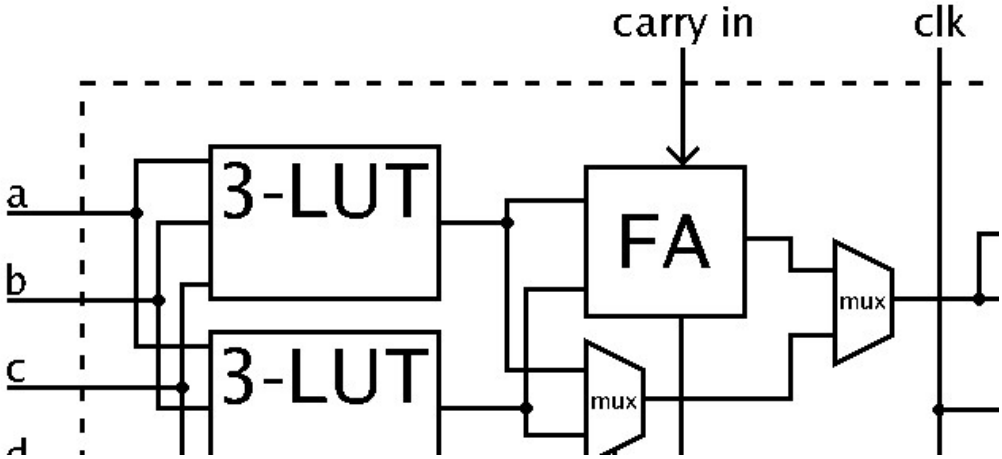


3 vstupy zodpovedajú 2^3=8 vstupným kombináciám uloženým vo vyhľadávacej tabuľke **LUT (Look Up Table)**
Ak je použitá **LUT**, klasická optimalizácia realizácie logickej funkcie stráca význam, keďže v tabuľke je možné umiestniť priamo **všetky hodnoty** realizovanej logickej funkcie.

Principiálna štruktúra základného stavebného bloku - logického elementu (LE, CLB, ...) v obvodoch FPGA

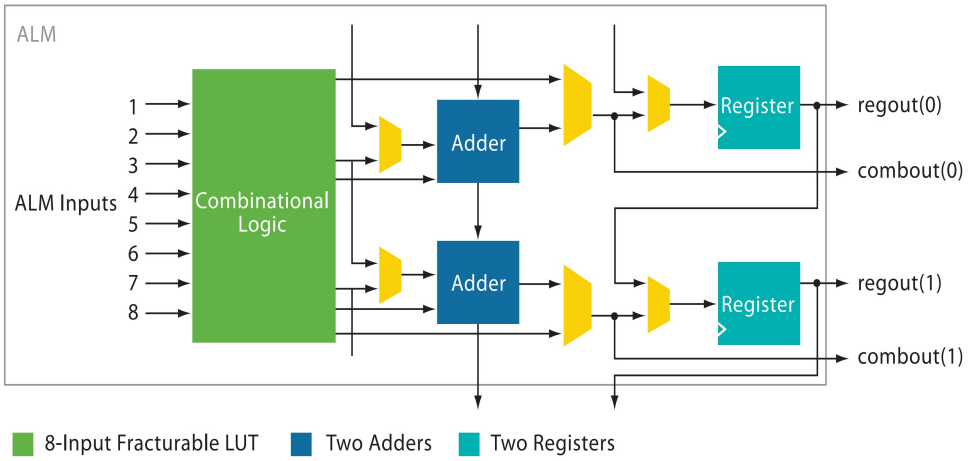


LE umožňuje realizovať aj kombinačnú logiku (s využitím **LUT**) a tiež sekvenčnú logiku s využitím synchrónneho preklápacieho obvodu (**FF**).

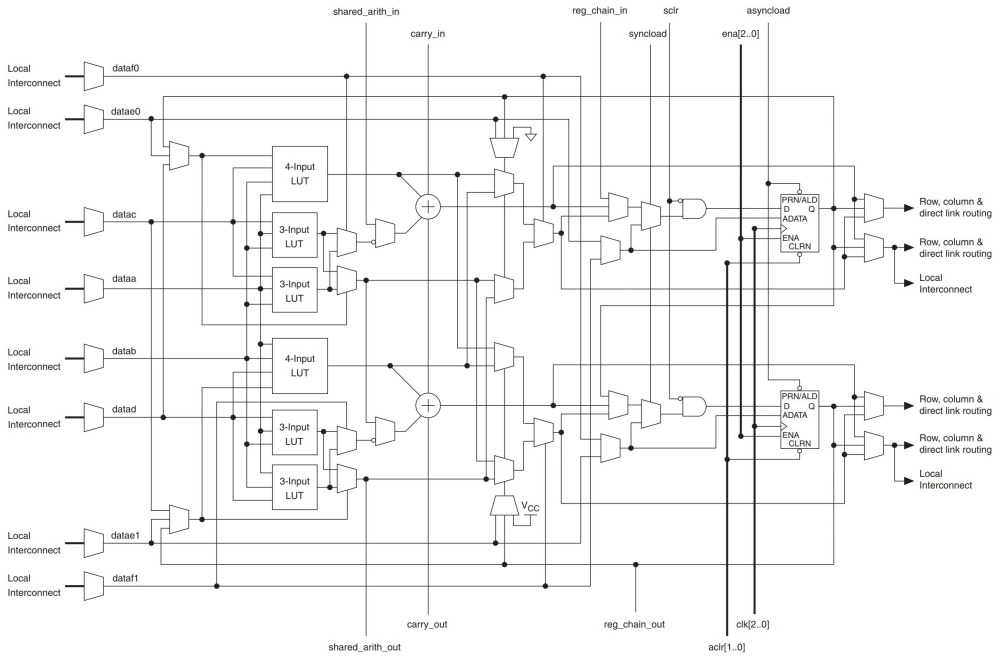


Zjednodušená schéma reálneho LE (**LUT** – Lookup Table, **FA** – Full Adder, **DFF** – D-type Flip-Flop). Štruktura sa mení v závislosti na rodine FPGA, všetky moderné FPGA však obsahujú minimálne vyššie uvedené bloky.

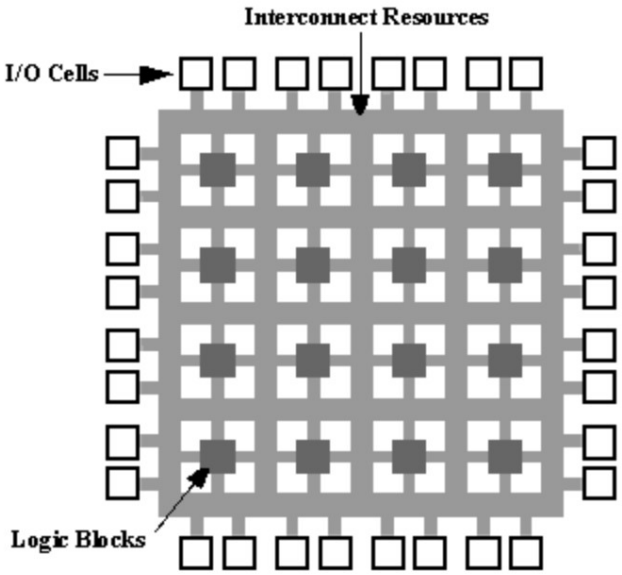
Príklad Adaptive Logic Module (ALM) v novších obvodoch Intel Altera FPGA



Príklad detailnej schémy ALM bloku v FPGA Intel (Altera Startix)



Principiálna štruktúra obvodu FPGA



FPGA interne obsahuje (okrem ďalších blokov ako hierarchické pamäte, vstavané násobičky, obvody PLL, ...) **veľké množstvo LE** (10-ky až 100-ky tisíc) a **prepojovacie pole**. V procese návrhu konfigurácie FPGA **prekladač** vytvára (hľadá a optimalizuje) vhodné **konfigurácie** všetkých použitých LE a tiež **vzájomné prepojenia** jednotlivých LE v cieľovom obvode FPGA.

Doplňujúce informácie:
https://vyvoj.hw.cz/zaklady-fpga-co-je-fpga-a-proc-je-pouzit.html

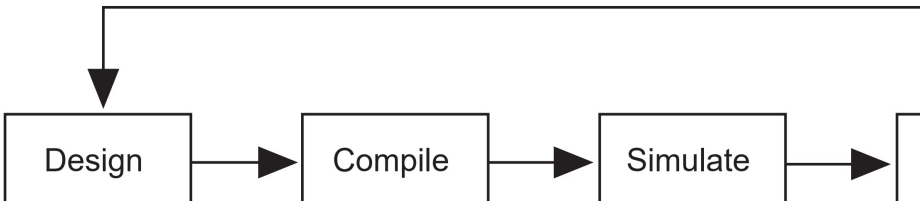
https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/nebojte-se-fpga.html

Príklad parametrov obvodov Cyclone IV FPGA
https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/pt/cyclone-iv-product-table.pdf

PRODUCT LINE		CYCLONE IV GX FPGAS[1]						
		EP4CGX15	EP4CGX22	EEP4CGX30	EP4CGX50	EP4CGX75	EP4CGX110	EP4CGX150
Resources	LEs (K)	14	21	29	50	74	109	150
	M9K memory blocks	60	84	120	278	462	666	720
	Embedded memory (Kb)	540	756	1,080	2,502	4,158	5,490	6,480
	18 x 18 multipliers	0	40	80	140	198	280	360
Clocks, Maximum I/O Pins, and Architectural Features	Global clock networks	20	20	20	30	30	30	30
	PLLs	3	4	4	8	8	8	8
	I/O voltage levels supported (V)							1.2, 1.
	I/O standards supported	LVTTL, LVCMOS, PCI, PCI-X, LVDS, mini-LVDS, RSDS, LVPECL, SSTL-18 (1 and II), SSTL-15 (I (I and II), Differential SSTL-15 (I and II), Differential SSTL-2 (I and II), Differential HSTL-						
	Emulated LVDS channels	9	40	40	73	73	139	139
	LVDS channels, 840 Mbps (receive/transmit)	7/7	14/14	14/14	49/49	49/49	59/59	59/59
	Transceiver count[2] (2.5 Gbps/3.124 Gbps)	2/0	2, 0 / 4, 0	4, 0 / 0, 4[3]	0, 8	0, 8	0, 8	0, 8
	PCIe hardened IP blocks (Gen1)	1	1	1	1	1	1	1
	Memory devices supported							DDR
Package Options and I/O Pins: General-Purpose I/O (GPIO) and Transceiver Count								
E144 pin[4] (22 mm, 0.5 mm pitch)		–	–	–	–	–	–	–
M164 pin (8 mm, 0.5 mm pitch)		–	–	–	–	–	–	–
M256 pin (9 mm, 0.5 mm pitch)		–	–	–	–	–	–	–
U256 pin (14 mm, 0.8 mm pitch)		–	–	–	–	–	–	–
U484 pin (19 mm, 0.8 mm pitch)		–	–	–	–	–	–	–
F169 pin (14 mm, 1.0 mm pitch)		72 2	72 2	72 2	–	–	–	–
F256 pin (17 mm, 1.0 mm pitch)		–	–	–	–	–	–	–
F324 pin (19 mm, 1.0 mm pitch)		–	150 4	150 4	–	–	–	–

Príklad pomerne starého typu FPGA z rodiny „low-cost“ FPGA firmy Intel

Postup návrhu FPGA obvodov



V rámci predmetu APS budeme realizovať len kroky
Návrh – s využitím VHDL
Kompilácia – s využitím Intel Quartus
Simulation – s využitím Altera-Modelsim

Konfigurovanie (slengovo často označované ako programovanie**)**
a **HW verifikácia** bude realizovaná vo voliteľných predmetoch.

Najvýznamnejší výrobcovia FPGA obvodov

Xilinx (od roku 2022 vo vlastníctve firmy **AMD**)
https://en.wikipedia.org/wiki/Xilinx
https://www.xilinx.com/

Altera (od roku 2015 vo vlastníctve firmy **Intel**)
https://en.wikipedia.org/wiki/Altera
www.intel.com/content/www/us/en/products/programmable.html

Microsemi (predtým Actel, od roku 2018 vo vlastníctve firmy **Microchip**)
https://en.wikipedia.org/wiki/Microsemi
https://www.microsemi.com/

Súčasný trend je vytváranie **SoC** (**System on Chip**), ktorý okrem konfigurovateľnej časti (bloky ALM – iní výrobcovia ich nazývaju inak, hierarchickýh pamätí, násobičiek, PLL obvodov, ...) obsahujú aj pevne **integrované CPU**, v súčasnosti najčastejšie na báze **ARM architektúry** (od jednoduchých ARM Cortex-M3 až po viac-jadrové ARM Cortex-A53). Alternatívou je aj možnosť vytvoriť **syntetické CPU** s využitím konfigurovateľných zdrojov FPGA (napr. ARM Cortex-M1, Nios, MicroBlaze, ...).

Soft core vs. **Hard core** – pozri napr.
https://www.uio.no/studier/emner/matnat/fys/FYS4220/h20/lecture-slides/embedded_softcore_vs_hardtcore.pdf